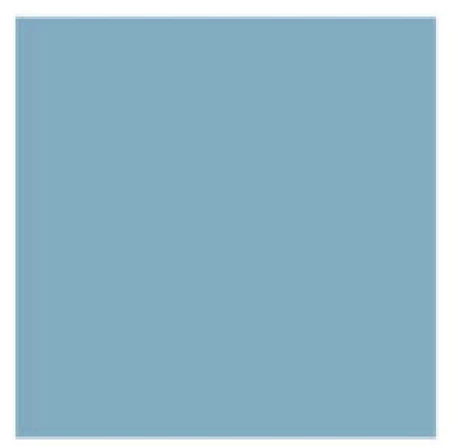
FRAMEWORK ATLAS

MANUAL DE USUARIO COMPONENTE LISTA PAGINADA EXTENDIDA

Versión 1.0

UNIDAD DE ARQUITECTURA DE SOFTWARE DE APLICACIONES





Agencia de
Informática y Comunicaciones
de la Comunidad de Madrid
icm

Hoja de Control

Título	Manual de usuario Componente Lista Paginada Extendida		
Documento de Referencia			
Responsable	Unidad de Arquitectura Software de Aplicaciones		
Versión	1.0	**Fecha Versión**	15/01/2015

Registro de Cambios

Versión	Causa del Cambio	Responsable del Cambio	Fecha
1.0	Versión inicial del documento	Arquitectura de Software	02/12/2014

Índice

1 INTRODUCCIÓN

Este documento contiene el manual de uso del componente Lista Paginada Extendida del framework Atlas. En él se incluye información sobre cómo utilizar dicho componente en una aplicación Web, así como información acerca de la configuración de los parámetros fundamentales del componente.

1.1 AUDIENCIA OBJETIVO

Este documento está orientado a toda aquella persona que esté desarrollando una aplicación Web basada en el Framework Atlas y necesite utilizar componentes de presentación en su aplicación Web.

1.2 CONOCIMIENTOS PREVIOS

Para un completo entendimiento del documento, el lector deberá tener conocimientos previos sobre las siguientes tecnologías:

- Java Server Faces (JSF)
- RichFaces
- Spring Framework
- Hibernate
- JavaScript
- CSS

Para saber más sobre dichas tecnologías, consultar el apartado de este documento, ENLACES RELACIONADOS.

2 DESCRIPCIÓN

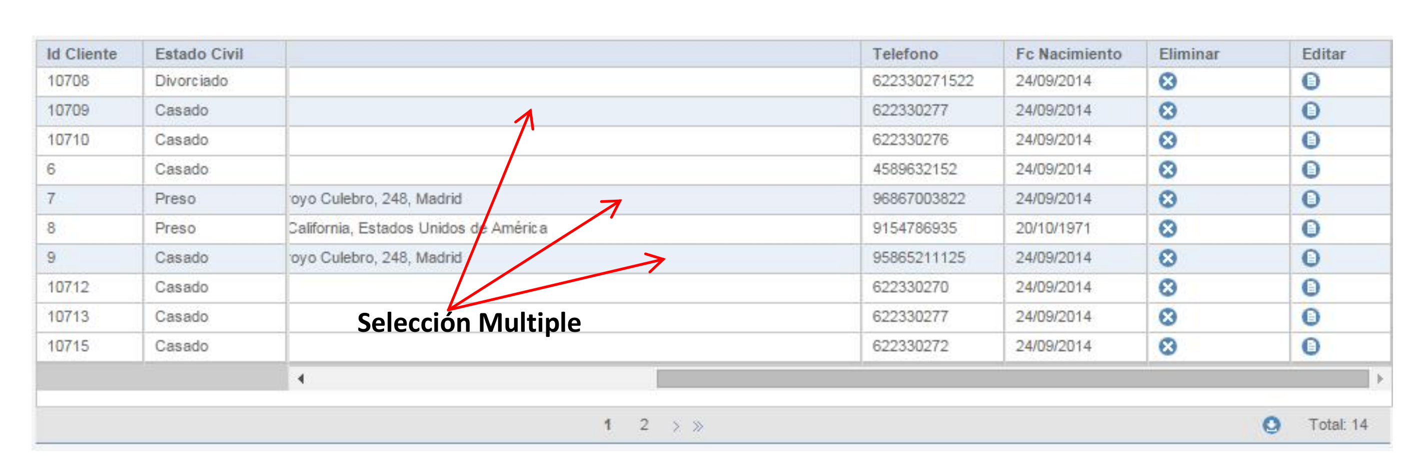
El componente Lista Paginada Extendida (*atlasc:listaPaginadaExtended*) esta construido tomando como base el componente Lista Paginada agregando nuevas funcionalidades como scroll vertical y horizontal, columnas fijas, selección multiple de filas, ancho inicial de las columnas y el poder arrastrar y soltar una columna para moverla a una nueva posición.

A continuación detallamos cada una de las nuevas caracteristicas:

- **Columnas Fijas**: mediante el atributo *frozenColumns* se puede establece el número de columnas que quedarán fijas en la tabla por lo que se podrá hacer scroll vertical con el resto de columnas de la tabla. A continuación se muestra un ejemplo de 2 columnas fijas:

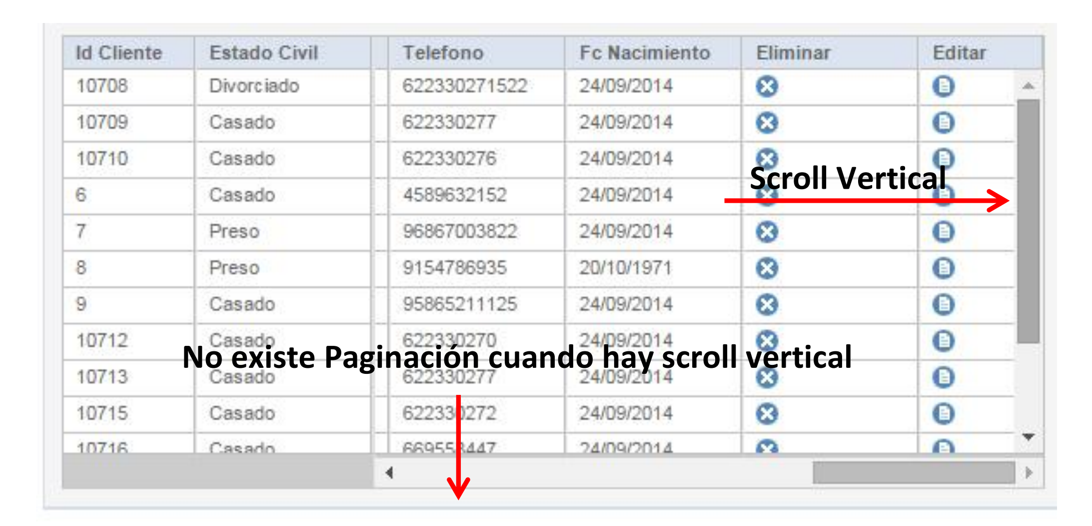
Id Cliente	Estado Civil	bre	Apellido 1
10708	Divorciado	re2	apellido2
10709	Casado	re3	apellido3
10710	Casado	re4	apellido4
6	Casado	Antonio	Jimenez
7	Preso	Vicente	Perez
8	Preso	n	Cordozar
9	Casado	Josefa	Jimenez
10712	Casado	re6	apellido6
10713	Casado	re7	apellido7
10715	Casado	re9	apellido9
10716	Casado	re10	apellido10

- **Selección multiple de filas**: mediante el atributo *selectionMode* se puede establecer el tipo de selección de las filas de la tabla, los posibles valores son: none(ninguna), single(solo una fila puede ser seleccionada), multiple(se usa en conjunto con las teclas Ctrl/Shift), multipleKeyboardFree (uso de clicks para multi selección):



Id Cliente	Estado Civil		Telefono	Fc Nacimiento	Eliminar	Editar
10708	Divorciado		622330271522	24/09/2014		
10709	Casado		622330277	24/09/2014		
10710	Casado		622330276	24/09/2014		
6	Casado		4589632152	24/09/2014		
7	Preso	oyo Culebro, 248, Madrid	96867003822	24/09/2014		
8	Preso	California, Estados Unidos de América	9154786935	20/10/1971		
9	Casado	oyo Culebro, 248, Madrid	95865211125	24/09/2014		
10712	Casado		622330270	24/09/2014		
10713	Casado		622330277	24/09/2014		
10715	Casado		622330272	24/09/2014		

- **Scroll Vertical**: mediante el atributo *clientRows* se establece el número de filas que se cargan mediante ajax cada vez que se baje el scroll solicitando más filas, si el número es más grande que el total de filas entonces se trae todas las filas en una sola llamada, cuando se dan estas condiciones no existen los botones de paginación puesto que el scroll hace que se traiga todos los registros sin paginar.



Id Cliente	Estado Civil	Telefono	Fc Nacimiento	Eliminar	Editar
10708	Divorciado	622330271522	24/09/2014		
10709	Casado	622330277	24/09/2014		
10710	Casado	622330276	24/09/2014		
6	Casado	4589632152	24/09/2014		
7	Preso	96867003822	24/09/2014		
8	Preso	9154786935	20/10/1971		
9	Casado	95865211125	24/09/2014		
10712	Casado	622330270	24/09/2014		
10713	Casado	622330277	24/09/2014		
10715	Casado	622330272	24/09/2014		

- **Ancho de columnas**: mediante el atributo *width* del componente *<atlas:columnaListaPaginada>* se puede establece el ancho inicial de las columnas de las tablas, este ancho se puede modificar en tiempo de ejecución usando el raton sobre las columnas de la tabla. Para modificar el ancho de la columna hay que poner el ratón a la derecha de columa hasta ver el icono ⇔ y luego arrastrar hasta que se obtenga el ancho deseado de la columna.
- **Reordenamiento de columnas**: el componente permite el arrastrar y soltar una columna bien hacia la derecha o izquierda hasta la posición deseada. En la siguiente figura se muestra un ejemplo donde la columna Apellido 2 se mueve hacia la posición que esta entre la columna Nombre y Apellido 1, se puede ver que haya una flecha hacia arriba de color azul que indica donde se va a poner la columna arrastrada.

Id Cliente	Estado Civil	Nombre	Apellido 1	Apellido 2
10708	Divorciado	nombre2	ape	segundoapellido2
10709	Casado	nombre3	apellic	segundoapellido3
10710	Casado	nombre4	apellido4	segundoapellido4
6	Casado	Jose Antonio	Jimenez	De laOsaccc4
7	Preso	Jose Vicente	Perez	Martinez4
8	Preso	Calvin	Cordozar	Broadus Jr.
9	Casado	Maria Josefa	Jimenez	De la olla
10712	Casado	nombre6	apellido6	segundoapellido6
10713	Casado	nombre7	apellido7	segundoapellido7
10715	Casado	nombre9	apellido9	segundoapellido9

1 2 › »

A continuación se muestra un ejemplo completo de la apariencia visual del componente:



Id Cliente	Estado Civil	2	Direccion	Telefono	Fc Nacimiento	Eliminar
7	Preso	4	Calle nº 4, Arroyo Culebro, 248, Madrid	96867003822	24/09/2014	
8	Preso	Jr.	Long Beach, California, Estados Unidos de América	9154786935	20/10/1971	
10708	Divorciado	apellido2		622330271522	24/09/2014	
13710	Deshauciado		Calle nº 4, Arroyo Culebro, 248, Madrid	6665556665	23/11/2014	
10709	Casado	apellido3		622330277	24/09/2014	
10712	Casado	apellido6		622330270	24/09/2014	
10715	Casado	apellido9		622330272	24/09/2014	
10713	Casado	apellido7		622330277	24/09/2014	
9	Casado		Calle nº 4, Arroyo Culebro, 248, Madrid	95865211125	24/09/2014	
10710	Casado	apellido4		622330276	24/09/2014	

1 2 › » Total: 14

3 DIFERENCIAS RESPECTO AL COMPONENTE atlas:listaPaginada

El componente **Lista Paginada Extendida** tiene las siguientes diferencias respecto al componente **listaPaginada**:

- No tiene la opción de cambiar en caliente el número de filas de la tabla, esto debido a que este componente cuando tiene scroll vertical se define un número de filas que se cargan por ajax y entra en conflicto con la carga en caliente
- La lista paginada Extendida **no se ajusta al ancho de la pantalla** como hace la lista paginada debido a tiene scroll horizontal y hay que darle un tamaño específico tanto al total de la tabla como a cada una de las columnas. Si a las columnas no se les pone tamaño tendrán por defecto un tamaño de 100 pixeles cada una
- La lista paginada Extendida no se ajusta su alto dependiendo del número de filas, como si hace el componente lista paginada. Con lo que habrá indicar un tamaño especifico para el alto de la lista paginada extendida.
- El namespace es diferente uno usa facelets atlas:listaPaginada y el otro usa composite atlasc:listaPaginadaExtended

4 INSTALACIÓN Y CONFIGURACIÓN

En este apartado se incluye información sobre la instalación y la configuración de este componente

4.1 INSTALACIÓN

El componente de lista paginada extendida ya viene instalado en el arquetipo web, incluido con el módulo de componentes visuales. Por este motivo no es necesaria una instalación adicional si se parte del arquetipo.

4.1.1 CONFIGURACIÓN

No es necesaria ninguna configuración adicional si se ha partido del arquetipo web.

5 USO

Una vez instalado el módulo de componentes puede procederse a su utilización. Para ello deben realizarse los pasos indicados en los siguientes apartados:

5.1 Paso 1: Definición del espacio de nombres de etiquetas de Atlas Composite

Es necesario crear un fichero xhtml y establecer la definición del espacio de nombres para las etiquetas de componentes de Atlas Composite. Un ejemplo de cabecera de fichero xhtml es la siguiente:

Cabecera de fichero xhtml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"
        xmlns:h="http://java.sun.com/jsf/html"
        xmlns:atlasc="http://atlas.core.componentes/jsf/composite"
        xmlns:a4j="http://richfaces.org/a4j"
        xmlns:rich="http://richfaces.org/rich">
```

5.2 Paso 2: Inserción en la página de la etiqueta de <atlasc:listaPaginadaExtended>

Se incluirá una etiqueta <atlasc:listaPaginadaExtended> que inserta el componente en la página.

A continuación se muestra un ejemplo

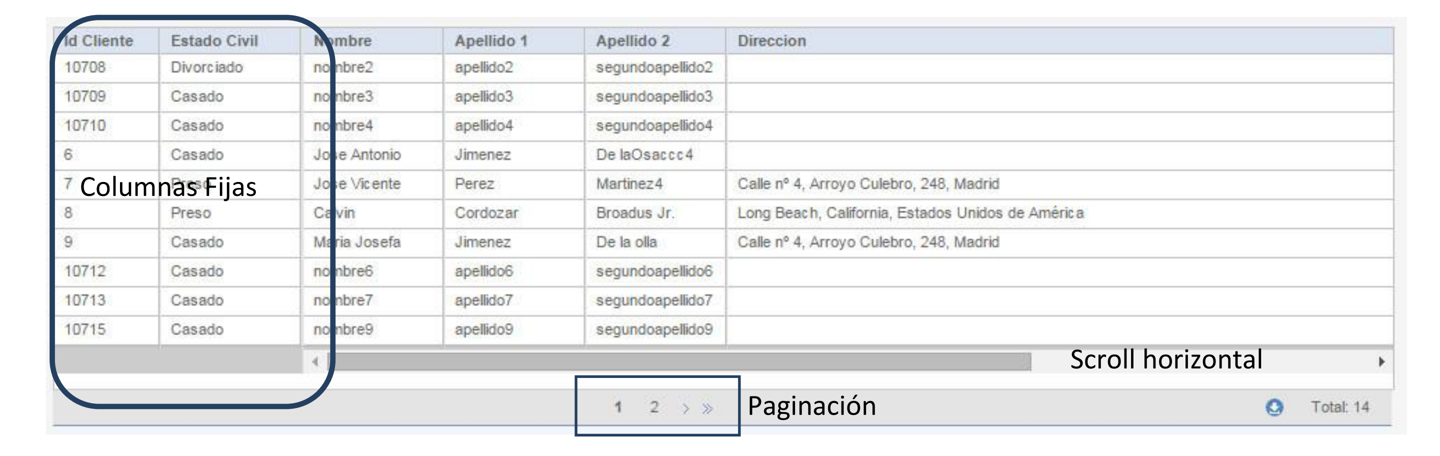
listaPaginadaExtendidaSample.xhtml

```
<atlasc:listaPaginadaExtended id="clienteTable"
  pageProvider="clienteBean.obtener"
  pageCountProvider="clienteBean.obtenerTotal"
  width="966px" height="260px"
  orderAndFilterProvider="clienteBean.getOrderAndFilter"
  sortAscending="#{clienteBean.orderAndFilter.ordenarAscendente}"
  sortColumn="#{clienteBean.orderAndFilter.columnaOrden}"
  filter="#{clienteBean.orderAndFilter.filtro}"
  export="true"
  frozenColumns="2"
  selectionMode="multiple"
  clientRows="0"
  rows="10">

<atlas:columnaListaPaginada id="columnaClienteIdCliente"
      for="clienteTable" label="#{bundle['table.clienteidcliente']}"
      orderAndFilterBean="#{clienteBean.orderAndFilter}" orderValue="idCliente"
      exportValue="idCliente" anchoColumna="75px">
       <ui:define name="value">
          <h:outputText id="outputTextClienteIdCliente"
```

```
            value="#{item.idCliente}">
            </h:outputText>
        </ui:define>
</atlas:columnaListaPaginada>
<atlas:columnaListaPaginada id="columnaClienteEstadoCivil"
      for="clienteTable" label="#{bundle['table.clienteestadocivil']}"
      orderAndFilterBean="#{clienteBean.orderAndFilter}"
      orderValue="estadoCivil.estadoCivil"
      exportValue="estadoCivil.textoListaValores">
       <ui:define name="value">
          <h:outputText id="outputTextClienteEstadoCivil"
          value="#{item.estadoCivil.textoListaValores}">
          </h:outputText>
       </ui:define>
</atlas:columnaListaPaginada>
…
```

A continuación se muestra un ejemplo de la lista paginada extendida sobre la tabla de clientes cuyo atributo más relevante es **frozenColumns=2**



Id Cliente	Estado Civil	Nombre	Apellido 1	Apellido 2	Direccion
10708	Divorciado	nombre2	apellido2	segundoapellido2	
10709	Casado	nombre3	apellido3	segundoapellido3	
10710	Casado	nombre4	apellido4	segundoapellido4	
6	Casado	Jose Antonio	Jimenez	De laOsaccc4	
7	[illegible]	Jose Vicente	Perez	Martinez4	Calle nº 4, Arroyo Culebro, 248, Madrid
8	Preso	Calvin	Cordozar	Broadus Jr.	Long Beach, California, Estados Unidos de América
9	Casado	Maria Josefa	Jimenez	De la olla	Calle nº 4, Arroyo Culebro, 248, Madrid
10712	Casado	nombre6	apellido6	segundoapellido6	
10713	Casado	nombre7	apellido7	segundoapellido7	
10715	Casado	nombre9	apellido9	segundoapellido9	

5.3 Lista paginada extendida con paginación o con scroll vertical

Si se quiere que la lista páginada extendida tenga un paginador como la lista paginada se tendrá que incluir el siguiente código

listaPaginadaExtendidaSample.xhtml

```
<atlasc:listaPaginadaExtended id="clienteTable"
  pageProvider="clienteBean.obtener"
  pageCountProvider="clienteBean.obtenerTotal"
  width="966px" height="260px"
  orderAndFilterProvider="clienteBean.getOrderAndFilter"
  sortAscending="#{clienteBean.orderAndFilter.ordenarAscendente}"
  sortColumn="#{clienteBean.orderAndFilter.columnaOrden}"
  filter="#{clienteBean.orderAndFilter.filtro}"
  export="true" frozenColumns="2" selectionMode="multiple"
  clientRows="0" rows="10">
      <atlas:columnaListaPaginada id="columnaClienteIdCliente"
..
```

Si se quiere que la lista páginada extendida no tenga paginador, pero tenga scroll vertical se deberá establecer la propiedad clientrows, que indica el número de filas que se cargan de golpe

AVISO
Esa opción se debe usar teniendo en cuenta que el número de filas a cargar no debería ser mayor de 20 ó 30 como máximo ya que puede repercutir negativamente en el rendimiento de la aplicación

listaPaginadaExtendidaSample.xhtml

```
<atlasc:listaPaginadaExtended id="clienteTable"
  pageProvider="clienteBean.obtener"
  pageCountProvider="clienteBean.obtenerTotal"
  width="966px" height="260px"
  orderAndFilterProvider="clienteBean.getOrderAndFilter"
  sortAscending="#{clienteBean.orderAndFilter.ordenarAscendente}"
  sortColumn="#{clienteBean.orderAndFilter.columnaOrden}"
  filter="#{clienteBean.orderAndFilter.filtro}"
  export="true" frozenColumns="2" selectionMode="multiple"
  clientRows="20" rows="0">
      <atlas:columnaListaPaginada id="columnaClienteIdCliente"
..
```



5.4 Atributos de la lista paginada extendida

ATRIBUTO	OBLIGATORIO	TIPO	DESCRIPCION
id	No	String	Identificador del componente. Recomendable especificarlo si se utilizan varias listas paginadas en la misma página.
clientRows	No	int	Usado para cambiar a AJAX lazy-loading mode. Specifica el número filas que se cargan en una llamada. Si es 0 entonces se cargan todas las filas
columnClasses	No	String	Estilos css para las columnas de la tabla
frozenColumns	No	int	Número de Columnas fijas de la tabla
headerClass	No	String	Estilos css para la cabecera de la tabla
onbeforeselectionchange	No	String	Método javascript que se llama antes de que cambie la selección de la fila.
onready	No	String	Método javascript que se llama despues que la tabla se ha inicializado.
onrowclick	No	String	Método javascript que se llama cuando se hace click sobre una fila de la tabla.
onrowdblclick	No	String	Método javascript que se llama cuando se hace doble click sobre una fila de la tabla
onrowkeydown	No	String	Método javascript ejecutado cuando se mantiene presionado una tecla sobre una fila de la tabla
onrowkeypress	No	String	Método javascript ejecutado cuando se presiona y suelta una tecla sobre una fila de la tabla
onrowkeyup	No	String	Método javascript ejecutado cuando se suelta una tecla sobre una fila de la tabla
onrowmousedown	No	String	Método javascript ejecutado cuando el boton del ratón se mantiene presionado sobre una fila de la tabla
onrowmousemove	No	String	Método javascript ejecutado cuando el puntero del ratón se mueve dentro de la fila de la tabla
onrowmouseout	No	String	Método javascript ejecutado cuando el puntero del ratón se mueve fuera de la fila de la tabla
onrowmouseover	No	String	Método javascript ejecutado cuando el puntero del ratón se mueve sobre la fila de la tabla

onrowmouseup	No	String	Método javascript ejecutado cuando el botón del ratón se suelta de la fila de la tabla
onselectionchange	No	String	Método javascript que se llama despues que se ha cambiado la selección de la fila
rowClasses	No	String	Asigna uno o mas estilos a las filas de la tabla
rows	No	int	Número de filas a mostrar en la tabla
selection	No	java.util.Collection	Colección de filas que se han seleccionado
selectionMode	No	org.richfaces.model.SelectionMode	Tipo de selección de las filas: none(ninguna), single(solo una fila puede ser seleccionada), multiple(se usa en conjunto con las teclas Ctrl/Shift), multipleKeyboardFree (uso de clicks para multi selección)
styleClass	No	String	Estilo de la lista paginada. Este campo es opcional, y por defecto vale 'tablaPijama'
pageProvider	Si	String	Método de backingbean para obtener los elementos de la página
pageCountProvider	Si	String	Método de backingbean para obtener el total de los elementos de la página
orderAndFilterProvider	Si	String	Método de backingbean para obtener el bean de filtrado de los elementos de la lista
sortAscending	Si	boolean	Boolean, para indicar si se ordena de forma ascendente por defecto
sortColumn	No	String	Atributo del ordenandFilterBean para guardar el nombre de columna de ordenación
filter	No	String	En el caso de que se utilice un filtrado para la query de la lista paginada, especifica el atributo donde se va a establecer el valor que representa el filtro
width	No	String	Ancho de la lista paginada, ejemplo “911px” **Es obligatorio indicar un ancho a la lista**
height	No	String	Altura de la lista paginada, ejemplo “206px” **Es obligatorio indicar un alto a la lista**
styleClassNavigator	No	String	Estilo del paginador de la tabla
widthNavigator	No	String	Anchura de la paginación del componente. Por defecto tiene el valor del campo width
leftNavigator	No	String	Distancia al margen izquierdo de la sección de

			paginación del componente. Campo opcional y valor por defecto '0px'
scrollerPanelColumns	No	String	Número columnas del datascroller
paginatorMaxPages	No	int	Atributo maxPages del dataScroller de Richfaces
renderIfSinglePage	No	boolean	Boolean. Muestra el datascroller si sólo hay una página. Su valor por defecto es true
fastStep	No	int	Atributo fastStep del dataScroller
export	No	boolean	Permite activar la exportación. Su valor por defecto es false
exportFile	No	String	Nombre que se dará al fichero de exportación, su valor por defecto es 'export'
exportLimit	No	int	Número máximo de filas a exportar

5.5 Nuevos atributos de la columna en la lista paginada extendida

ATRIBUTO	OBLIGATORIO	TIPO	DESCRIPCION
anchoColumna	No Por defecto 100px	String	Ancho en pixeles de cada columna

5.6 RECOMENDACIONES Y BUENAS PRÁCTICAS

5.7 EJEMPLO DE USO

Se puede ver 2 ejemplos de dicho componente en la aplicación de componentes (a partir de la versión 1.2.8) en la siguiente secuencia de navegación:
Inicio > Paginación de Resultados > Lista Paginada Extendida
Inicio > Paginación de Resultados > Lista Paginada Extendida con scroll

6 PREGUNTAS MÁS FRECUENTES

En este apartado se incluyen una lista de preguntas más frecuentes sobre el componente.

Pregunta: ¿Dónde puedo encontrar información general sobre los componentes?
Respuesta: En la aplicación de demostración de los componentes del Framework Atlas

Pregunta: ¿ Cómo se ha implementado el componente en su capa de presentación?
Respuesta: Se ha utilizado la tecnología Composite de JSF, Facelets, JavaScript y CSS

Pregunta: ¿Se puede usar una columna personalizada con este componente? Por ejemplo, ¿se puede introducir una columna con iconos o botones?
Respuesta: Si, se puede utilizando el atributo define de facelets. El componente columnaListaPaginada permite personalizar el contenido que se muestre. Por defecto se muestra un outputText, pero se puede utilizar cualquier otro componente como por ejemplo un commandLink. En la sección 5.2 hay un ejemplo de este caso

Pregunta: ¿ Cómo puedo modificar los estilos del componente?
Respuesta: Mediante los atributos de estilo del componente. Para más información consultar la tabla de atributos en la sección de Uso del componente

7 ENLACES RELACIONADOS

Producto	URL	Versión
RichFaces	http://docs.jboss.org/richfaces/4.3.X/4.3.1.Final/Developer_Guide/en-US/html_single/	4.3.1.Final
JSF	http://docs.oracle.com/javaee/6/javaserverfaces/2.1/docs/vdldocs/facelets/index.html	2.1
MyFaces Core	http://myfaces.apache.org/core21/index.html	2.1
JavaScript	https://developer.mozilla.org/es/docs/Web/JavaScript	
CSS	http://www.w3schools.com/css/default.asp	